

Identifying Food Fraud using Blockchain

Hoi Wen Leung, Adriane Chapman^a and Nawfal F. Fadhel^b

School of Electronics and Computer Science, University of Southampton, Southampton, U.K.

Keywords: Food Fraud, Food Supply Chain, Blockchain, Consensus Algorithm.

Abstract: Cross-contamination, counterfeit ingredients, false packaging, and labelling are all issues that contribute to food fraud which is a major concern undermining the integrity of the food supply chain and consumers health. Therefore, there is a need for an on-demand traceable, transparent food supply chain. This is a universal problem and blockchain presents itself as a means to maintain traceable, transparent food supply. This paper presents an innovative consensus algorithm and simulates the usage of it to identify the precision and recall of fraudulent food detection. This protocol aims to solve the issue of malicious leader node selection in common voting-based consensus protocols while achieving efficiency. Thus, providing a single version of truth for foods in a long food supply chain, preventing information asymmetries.


1 INTRODUCTION


With the demand for food product transparency on the rise, existing product traceability systems are not sufficient to maintain an immediate and reliable food analysis(Flari et al., 2014). This demand is mainly due to the increasing number of food fraud cases and incidents, such as the horse meat scandal and organic food counterfeit. These scandals jeopardise the credibility and consumer's trust of the food supply chain, thus introducing reputational risk and brand damage(Kamath, 2018). Upon investigating the scandals, one of the fundamental issues identified in the current food supply chain system is the serious delays in tracing food products(Litke et al., 2019). This is due to the complexity and fragmentation of the food supply chain. However, the adoption of industry 4.0 can evolutionise the food supply chain with the integration of technologies. It aims to target food safety and security in the hope of providing successful food supply chain management(Ojo et al., 2018).

A contributing factor to food fraud is the limited peer-to-peer transparency resulting from information asymmetries(Flari et al., 2014). In response to achieving end-to-end visibility along the food supply chain, many companies turn to solutions involving blockchain. In addition to its functionality as transparency in cryptocurrencies, many existing use cases that utilise blockchain to manage supply chain

logistics and monitor food product flow(Verhoeven et al., 2018). For instance, IBM and Walmart utilise blockchain to create a data exchange platform to achieve food provenance(Kamath, 2018). A startup called Bext360 also demonstrates a mindful use of this technology by binding data to bags of coffee to improve traceability(Verhoeven et al., 2018).

Although the current blockchain solutions in the food supply chain offer product traceability and better supply chain management(Verhoeven et al., 2018), there is limited research on identifying food fraud to safeguard the food supply chain. In this work, we analyse protocol for detecting food fraud. One problem with the current consensus protocols used in blockchain is that they are designed mainly for cryptocurrencies to validate transactions, such as Proof-Of-Work. This method allows a high level of trust to be developed between participants at the cost of high energy consumption overhead. Another popular protocol within the cryptocurrencies domain is Proof-Of-Stake, though less computationally expensive, it has the flaw of the "Nothing at Stake" problem, so blockchain participants can validate block transactions without opportunity cost(Alzahrani and Bulusu, 2020). Practical Byzantine Fault Tolerance is a voting-based consensus algorithm where a primary node is responsible for publishing blocks and their consensus(Litke et al., 2019), this suggests a major weakness if this leader node is malicious which in turn hinders the data integrity of the blockchain. For instance, the attacker aims to manipulate the fre-

^a  <https://orcid.org/0000-0002-3814-2587>

^b  <https://orcid.org/0000-0002-1129-5217>

quency of them being the leader in order to impact the validation outcome(Deirmentzoglou et al., 2019). Despite the decentralisation, transparency and security that blockchain provides, there is a need for a more suitable consensus protocol for food supply chain application(Litke et al., 2019) as well as overcoming the aforementioned vulnerabilities from other protocols.

To improve the detection of ingenuine products within the food supply chain and provide a single version of truth for the food products, we propose a new voting-based consensus protocol where every intermediary on the food supply chain can vote for the truthfulness of the transaction block. With the aim to prevent the existence of a single malicious leader node, multiple node selection stages are carried out which choose validators and validation master randomly to validate blockchain participants' votes and block outcome respectively. In this work, we simulate the usage of the blockchain consensus protocol in the food supply chain to test the precision and recall of fraudulent food detection, as well as the prevention of various types of attacks. Our contributions include:

1. Identified a novel use case for blockchain to protect the food supply chain. See Section 3.
2. Created a new voting-based consensus protocol that allows food supply intermediaries to vote for the block truthfulness, using multiple node selection stages to choose validators and validation masters randomly. See Section 4.
3. The precision and recall of fraud detection highlighted by this approach, as well as the attacks prevented. See Section 5.2.
4. Analysed the use of the proposed protocol for fraud identification through simulation. See Section 5.3.

2 RELATED WORK

The problem of lack of transparency in the food supply chain has been an ongoing issue. It is clear that adopting blockchain technology brings substantial improvements in providing traceability information(Verhoeven et al., 2018; Kamath, 2018; Shevchuk, 2019). Kamble et al. perform a quantitative study through a combined methodology of Interpretive Structural Modelling (ISM) and Decision-making Trial and Evaluation Laboratory (DEMATEL) to evaluate the links between the enablers. In particular, they introduce the importance for a consensus mechanism to assure information authenticity. Their result demonstrates blockchain's potential

in enhancing traceability, auditability and provenance in supply chains(Kamble et al., 2020).

Block-Supply is presented by Alzahrani and Bulusu to target the problem of counterfeit goods. They made use of the decentralised approach to track products to detect fraudulent activities. A new consensus protocol based on Tendermint is also introduced. This protocol employs the concept of voting-based consensus protocols to eliminate the need for existing mining which is commonly used in Blockchain(Alzahrani and Bulusu, 2020). Another consensus protocol, FireLedger examines the trade-off between finality and latency when achieving agreement on a permissioned blockchain. Although FireLedger proposed protocol is designed for cryptocurrencies applications, it emphasised the requirements of validity, agreement and termination which are also identified in this paper(Buchnik and Friedman, 2019).

Fadhel et al. utilises threat modelling to analyse how the possible security issues of the Internet-of-Things(IoT) can put the proposed blockchain-based smart grid at risk. This approach provides information of the involved parties and their activities so as to assess the quality and reliability of the suggested architecture(Fadhel et al., 2019). Similarly, our attack vectors are modelled using agents with respect to the food supply chain entities and their behaviours. This approach enables us to evaluate the reliability of the proposed protocol.

3 FOOD SUPPLY CHAIN USE CASE

The food supply chain is a complex network that describes the process of food production from its origin to the final product. As food products have vastly different production procedures due to their distinctiveness(Shevchuk, 2019), for the purpose of understanding how the proposed design impact fraud detection, we narrow the focus to the milk supply chain. It is a common practice for fraudulent actors to add foreign substances to the milk with the aim to increase product quantity for illegal profits(Hong et al., 2017).

According to European Union Food Legislation, the food supply chain is composed of production, processing, transport, distribution and supply¹. This is illustrated in Fig. 1 in the context of milk production, and the following stakeholders and their responsibilities are considered:

¹<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=legissum:f80501>

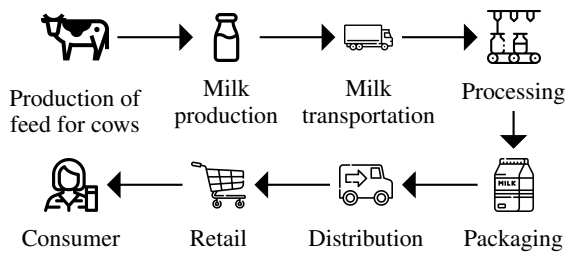


Figure 1: Milk supply chain.

1. **Farmers:** Responsible for converting raw material (cows) to a product (milk) used as input to the system.
2. **Processors:** In charge of transforming raw milk. They also manage the packaging of the product. They have the right to modify product details in an authorised way.
3. **Distributors:** Organises milk product transportation to its destination.
4. **Retailers:** The receiving ends of the distribution, such as supermarkets, restaurants etc.

3.1 Attack Vectors

A blockchain food traceability system can be utilised further than product flow monitoring to fraudulent product prevention across the food supply chain. Thus, it is important to outline the various existing weaknesses and threats on the blockchain application and the food supply chain. Table 1 describes the most concerning attacks in the food supply chain.

Due to the possible existence of the aforementioned attacks in the blockchain network, trustful cooperation must be established between the nodes. Therefore, the role of consensus protocol is to enable users to trust the blockchain output and its ability in detecting fraud(Hammi et al., 2018).

Table 1: Attack vectors specific to blockchain application and food supply chain.

| Attack | Description |
|---------------------|--|
| Modification attack | The malicious actor tampers the product details of a transaction block in an unauthorised way(Alzahrani and Bulusu, 2020). |
| Spoofing attack | The attacker disguises as legitimate food supply chain actor and participate in block validation process with the aim to disrupt the blockchain's operation(Hammi et al., 2018). |

3.2 Requirements

For the purpose of defining the functionality of the consensus protocol, a bottom-up approach is employed to elicitate the requirements from use cases. This method allows emphasis on the food supply chain actor's goal of the proposed solution by covering both the leader node selection and voting mechanism. All stakeholders are considered as primary actors, and the following summarises the use cases:

1. **Record Transaction:** Once the ownership of the milk product is transferred, the receiving actor scans the product detail and proposes a block containing the transaction and await for validation.
2. **Local Block Validation:** The blockchain participants receive a request to validate the newly proposed block with their copy of the blockchain, and decide on the genuinity of the block.
3. **Global Block Validation:** Validation leaders receive votes of the block from all live participants and validate the votes in several stages and decide on a final outcome to commit or abort the block.

Based on the use cases, functional requirements are derived and outlined below. They describe the consensus protocol's need for users to achieve a desired result(Hammi et al., 2018):

1. **Termination:** The consensus process eventually halts when all valid nodes reach an outcome.
2. **Total Ordering:** The sequence of votes and validation requests are processed in the correct order.
3. **Agreement:** The block on the blockchain must be globally recognised through consensus from all valid nodes.
4. **Validity:** If all valid nodes validate the same input block, then all values voted must be the same.
5. **Integrity:** All honest nodes should have the same shared state containing the agreed blocks.
6. **Zero-knowledge:** All nodes should have no knowledge of other nodes' votes and vote independently. Also, the nodes should not be able to predict the identity of leader nodes.
7. **Randomness:** Each validator is mapped to a validation master randomly and fairly.

4 APPROACH

As the aim is to address food fraud detection in the food supply chain, such consensus protocol should

solve the general block transaction agreement problem with minimal energy consumption (Litke et al., 2019). Hence, in this section, we propose a novel consensus mechanism by considering the benefits of the existing techniques and overcoming their weaknesses.

4.1 Key Components

During the process of executing the consensus protocol, each node takes on different duties. Thus, the following roles are considered:

1. **Proposer:** This node proposes a block recording the current transaction and broadcasts it for validation. There is only one proposing node for each consensus protocol execution.
2. **Voter:** Voting node performs local validation of the proposed block and decide the block's validity. All nodes except the proposing node are voters.
3. **Validator:** Validating node receives votes from all voters and return an outcome for the block. The number of validating nodes depends on the total number of nodes.
4. **Validation Master:** This node is responsible for performing the random selection of validation nodes. The master node also acts as a safety net to verify the outcome generated by validators.

Once consensus of the block is achieved successfully, it represents a valid milk product transaction on the blockchain network. We also redefined the block structure slightly for it to be suitable for blockchain in the food supply chain. A valid block consists of two main parts, the block header (B_{header}) and the block data (B_{data}). The block follows a structure:

$$Block_{valid} = B_{header} | B_{data} \quad (1)$$

$$B_{header} = Hash_{cur} | Hash_{prev} | Timestamp \quad (2)$$

$$B_{data} = P_{data} | Role | Sig | Address_s | Address_d \quad (3)$$

$$P_{data} = ID | Name | Origin | Farmer | Ingredients \quad (4)$$

The blockchain is ensured to be well-chained in the correct order using hashes of the current ($Hash_{cur}$) and previous ($Hash_{prev}$) blocks. The timestamp guarantees data immutability by preventing blockchain manipulation, thus enhancing information completeness. B_{data} contains the product data (P_{data}) which has fields such as *Origin* and *Farmer* to allow food to be traced from farm-to-fork. *Role* indicates the type of food supply chain actors and their modify rights of the product. This provides the ability for the protocol to detect modification attacks if block's fields are modified in an unauthorised way. In addition, the digital signature Sig is an essential part of blockchain. It is

generated using the cryptographic algorithm, ECDSA which does not require much computational power. The addresses of source and destination allows monitoring of the product's workflow and assure traceability.

4.2 Proposed Consensus Protocol

The protocol consists of three phases: a local validation phase where voters determine the truthfulness of a transaction block, a selection phase where validating nodes are chosen in a randomised and dynamic way, and a consensus phase where all participating nodes' votes are considered collectively to determine the validity of the proposed block.

For the protocol to provide Byzantine fault tolerance, it is assumed that the blockchain network has two-third of non-malicious nodes. We also introduced a timeout consistently throughout the protocol. This timeout is relative to the size of the network to guarantee the liveness of the protocol and allow the detection of any malfunctioned or disconnected nodes.

4.2.1 Local Validation Phase

This phase is executed by all voters when after a node receives the food product and proposes a block. Each voter individually validates the proposing block and generates a vote accordingly.

To understand what criteria needs to be checked whilst validating a transaction, we lay out how the genesis block is created with an example. Consider the first stage of the milk supply chain, when the farmer produces raw milk from cows. Once the product is created, there is a digital label attached which represents the physical form of the product. The farmer scans this label storing information about the product, acting as the preliminary input into the blockchain system. The system then creates a proposing block with content listed in Section 4.1.

The proposed block is digitally signed with Sig . In this case, ECDSA generates Sig by utilising the public (PU_s) and private keys (PR_s) from the source which is the farmer along with the public key (PU_d) from the destination node which is the processor. The product data (P_{data}) as follows:

$$Sig = Sign_{PR_s}(PU_s | PU_d | P_{data}) \quad (5)$$

As the farmer is the proposer of the genesis block, their digital signature is attached to the block. Henceforth, the ECDSA-generated digital signature can act as a means to verify the authenticity of the block content efficiently.

The local validation process is explained in Algorithm 1. This validation detects modification attacks since product details and the block's digital signature are considered. The proposed block ($Block_i$) is only considered to be valid if the following are satisfied:

1. Given B_{data} of $Block_i$, the digital signature is verified using PU_s .
2. P_{data} on $Block_i$ is the same as the ones on the last block from the node's copy of the blockchain (BC).
3. P_{data} is modified only in an authorised way.
4. $Hash_{cur}$ of $Block_i$ is calculated correctly.
5. $Hash_{prev}$ of $Block_i$ is the same as $Hash_{cur}$ of $Block_{i-1}$.

Algorithm 1: Block validation.

```

Require:  $Block_i \neq \text{NULL}$ 
if  $Block_i \neq \text{calculateHash}()$  OR  $\text{!verifySignature}(\text{Sig})$  OR  $Block_{i-1}.\text{hash} \neq \text{previousHash}$  then
  return false
end if
if  $\text{modifyRight} == \text{false}$  then
  if  $ProductD_{i-1} \neq ProductD_i$  then
    return false
  end if
end if
return true

```

4.2.2 Selection Phase

This phase occurs after local validation is completed. It highlights how the validation master and validators are selected in a random and fair manner. This randomised node selection prevents malicious nodes from predicting who the selected node are.

The algorithm presented in Algorithm 2 shows the validation master selection process. To begin, each node has the probability of $\frac{1}{n}$ from being selected as validation master, with n being total number of nodes $- 1$. All non-proposing nodes generate a random selection value (SV_m). Additionally, the upper limit of the selection value range can be changed to adjust the randomness.

To universally decide on the validation masters, each node requests other's selection value (SV_n) and executes a comparison between all these values and a randomly generated deciding value (DV), so the resulting two nodes with the closest SV_m are chosen to be validation masters. This selection process is performed by all non-proposing nodes to ensure correctness of the chosen master, so no malicious nodes can claim the role of validation masters.

Algorithm 2: Validation master selection.

```

 $n \leftarrow \text{total no. of nodes} - 1$ 
 $\text{lim} \leftarrow \text{static random no.}$ 
 $SV_m \leftarrow \text{random no. up to lim}$ 
 $\text{threshold} \leftarrow n \times \frac{2}{3}$ 
 $\text{timeout} \leftarrow n \times 10 \text{ seconds}$ 
for  $\text{round} \leftarrow 1$  to 3 do
  Multicast(ASK 0)  $\rightarrow$  all voters
  while  $\text{!timeout}$  do
     $\text{valueList.insert}(\text{received } SV_n)$ 
  end while
  if  $\text{valueList.size} == n$  OR  $\text{valueList.size} \geq \text{threshold}$  then
    Break
  end if
   $\text{round}++$ 
end for
if  $\text{round} > 3$  OR  $\text{valueList.size} < \text{threshold}$  then
  Exit
end if
 $DV \leftarrow \text{lim} \div \text{random no.}$ 
 $\text{valueList.map}(SV_n - DV)$ 
 $\text{valueList.sort}()$ 
 $\text{master1} \leftarrow \text{valueList}[0]$ 
 $\text{master2} \leftarrow \text{valueList}[1]$ 

```

After successful mapping of validation masters to a proposer, the validators selection is carried out. The masters share the workload of selecting the validators and handling the communication between them. Consequently, each master selects half of the required number of validators. The number of validators was determined to be $\log(n)$, with four nodes being the minimum number of validators required to tolerate single Byzantine fault (Alzahrani and Bulusu, 2020). The size given by the logarithm allows communication overhead to be optimised while making sure spoofing attack can be protected against. This process is similar to the validation master selection but without the use of selection and decision values.

It is important to note that the visibility of communication between validators and validation master is not open to all milk supply chain participants, so only the validators have knowledge of who the master nodes are. Furthermore, each master does not share the same validators as they execute the validators selection algorithm independently to prevent collusion of the selection. Therefore, this satisfies the requirements of zero-knowledge and randomness.

4.2.3 Consensus Phase

When validation masters and validators are selected, the protocol is ready to enter the consensus phase.

The following details the process from the milk supply chain actor receiving the product, to validation masters considering all the votes to decide collectively whether the proposed block should be on the ledger. It is worth mentioning that all types of messages are signed by the private key of the sender, and each message has a timestamped ID attached to preserve total ordering.

1. A node receives the milk product from the previous actor of the milk supply chain. This node then becomes a proposer and proposes $Block_i$.
2. The proposer broadcasts $Block_i$ to the blockchain network, so each voter executes the local validation algorithm and vote for $Block_i$. This stage should prove validity of the protocol.
3. Voters prepare for validation master selection by generating SV_m . Then the network uniformly and randomly selects two masters. This SV_n exchange also acts as acknowledgements from other nodes, so each voter keeps track of non-Byzantine nodes.
4. The validation masters perform handshakes with each other to verify their identities.
5. The validation master each selects $\frac{\log(n)}{2}$ validators randomly. The masters multicasts a request for acknowledgement to the chosen validators.
6. After confirmed selection of the validators, each validator requests vote from all voters and decides on an outcome if there are at least $\frac{2n}{3}$ responses. $Block_i$ is only determined to be genuine or ingenuine if all received votes align. Otherwise, the outcome is inconclusive. If the number of rounds is still within the limit, validators request the voters to perform local validation again and send the new votes. However, if there are $\frac{n}{3}$ or less nodes with votes different to the rest of the voters, these voters are concluded to be malicious. It is worth noting that the use of rounds limit avoids never-ending loops and ensures the protocol terminates.
7. When each validator has decided on an outcome, the validation masters are informed of the outcome by their chosen validators. If the validators consistently give mismatched responses, this signals a possible existence of malicious validators. A master-outcome is then determined by each master if each receives the same outcome from their set of validators.
8. Finally the validation masters communicate with each other to check if their master-outcome matches. A master with different master-outcome signals a possible malicious validation master node. Otherwise, the block is committed to the ledger or aborted accordingly.

5 EVALUATION

This section describes the simulation used to evaluate the performance of our consensus protocol. Specifically, we considered the attack vectors mentioned in Section 3 to determine the protocol's ability in detecting food fraud on the blockchain application.

5.1 Simulation Setup

Due to the lack of platform to simulate blockchain, we implemented a basic blockchain network in Java. Also, various cryptography functions from Bouncy Castle was used to generate keys pair, calculate hashes and apply digital signatures². This blockchain is modelled to be a permissioned because confidentiality and privacy of the food supply chain actors need to be protected against their competitors.

Test scenario cases were defined based on the attack vectors. These cases are used as input files for the simulation of the food supply chain. The actors are modelled as agents in the simulation with $\frac{n}{3}$ or less actors performing malicious behaviours to mimic the possible threats. The cases are defined in a way that all the aforementioned attacks are covered and with various levels of difficulties to be detected. Correspondingly, the simulation of the attack model are introduced in Table 2.

Table 2: Attack modelling overview.

| Attack | How it is modelled | Case count |
|---------------------|---|------------|
| Modification attack | Malicious node alters fields of the products before proposing the a new block. | 36 |
| Spoofing attack | Attacker act as genuine nodes and perform activities to disrupt the consensus protocol. | 12 |

As modification attack defines the illegitimacy of the new block, so all honest nodes should vote the block as invalid to maintain protocol's validity. As for spoofing attack, the types of activities depends on the node's role during the consensus protocol. Consider an attacker posing as a genuine voter to disorganise the consensus protocol, this node would misbehave in the voting process. For instance, they would vote valid regardless of the result from the local validation, threatening the data integrity and validity of the blockchain network. If the node is a malicious validator, it determines the block outcome to be valid

²<https://www.bouncycastle.org/>

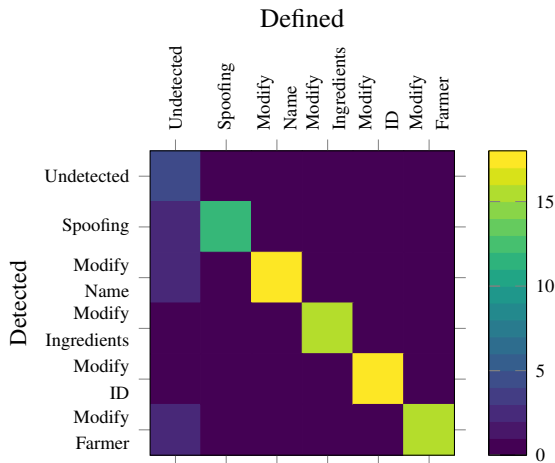


Figure 2: Error matrix of consensus protocol.

without considering any votes received from the voters. Similarly, for a corrupted validation master, they would set the master-outcome to be valid and disregard the outcome generated from validators. As a result, we have modelled dishonest nodes to always determine the proposed to be valid regardless of the actual authenticity of the block.

5.2 Detection of Fraud

To answer how reliable the consensus protocol is in detecting food fraud, we evaluate this based on the output generated from running the test cases. There are labels associated with the test cases which indicate if a defined test scenarios case is no fraud, modification attack or spoofing attack. Therefore, our aim is to produce the same label as the defined label after the simulations.

The result generated from the simulations after running all the test cases is visualised using an error matrix in Figure 2. This two-dimensional contingency table shows to show whether the correct types of food fraud are detected correctly.

The diagonal elements show correct fraudulent scenarios detected by the system, whereas the other cells are missorted entries. It is obvious that some cases are identified as having no threats incorrectly. These cases occur because there are more than $\frac{1}{3}$ malicious nodes which the model is not designed to tolerate. Otherwise, the shaded diagonal elements provide evidence that the model is very effective in identifying food fraud cases.

To interpret the results better, we used the performance metrics that can be derived from the error matrix. These metrics are precision and recall. Precision assesses the proportion of correctly detected fraudu-

lent scenarios cases amongst the fraud detected by the protocol. It is a useful indicator of how precise the consensus protocol is out of the detected fraud. Precision calculates the number of true-positives (TP) over the the sum of TP and false-positives (FP), across fraud of all labels. The formula for precision is given in Equation 6.

On the other hand Equation 7 presents recall, this is used because of the high cost of false-negatives (FN). In the context of food fraud, this cost is the cost when a fraud is mislabelled as non-fraud. Recall looks at the proportion of fraudulent scenarios that are actually detected by the protocol over the number of defined fraudulent scenarios, given by TP plus FN .

$$Precision = \frac{\sum_{i=1}^k \frac{TP_i}{TN_i + FP_i}}{k} \quad (6)$$

$$Recall = \frac{\sum_{i=1}^k \frac{TP_i}{TP_i + FN_i}}{k} \quad (7)$$

The values of precision and recall were computed to be 0.941 and 0.933 respectively. The precision means that 94.1% of the detected fraudulent cases are actually fraudulent as defined by the test scenarios. This indicates a high frequency of the model being correct when fraud is detected. For recall, 93.3% of the defined fraud scenarios are able to be detected by the system correctly. Thus, recall demonstrates the implemented protocol's ability in returning most of the defined scenarios. Subsequently, the high values indicate the consensus protocol's good performance in detecting modification and spoofing attacks.

5.3 Analysis of Available Attack Vectors

As blockchain is a trustless network, security is an essential factor. Hence, we provide analysis of the possible attacks presented in Table 2 and how our proposed protocol mitigate them.

In the proposed protocol, the local validation is mainly responsible for detecting modification attack which is the most vulnerable during block's transaction data generation. Therefore, the placement of the local validation algorithm enables honest voters to reject the ingenuine block, improving the detection of modification attack. Besides, as each block utilises digital signature, this guarantees block originality. If anyone attempts to modify the content, the signature will be invalidated and the attack is detected.

Our consensus protocol is also protected against spoofing attacks. As each node can only have a single identity and the feature of permissioned blockchain only allows approved nodes to join the network, the attacker cannot disguise as a genuine blockchain node

without their identity authenticated. Furthermore, each legitimate node holds a public-private key pair, so the attacker cannot spoof another node's identity since they do not have the required private key.

6 CONCLUSIONS

In this paper we presented a novel consensus protocol suitable for the food supply chain. We utilised validation, selection and consensus mechanisms to achieve validity, randomness and agreement. This protocol can detect modification, spoofing attacks on blockchain used in the food supply chain. Our simulation shows that the protocol is very capable of tolerating such attacks, but scalability and latency have yet to be evaluated. Additionally, further work are required to assess the consensus protocol's performance in detecting other attacks. Nevertheless, this proves the protocol's reliability in detecting food fraud.

7 FUTURE WORK

Apart from the concerned attack vectors suggested in Section 3, there are other security threats that our protocol can be protected against. Therefore, these attacks will require further modelling and evaluation in our future work.

In mining-based consensus protocol, bribery attack is introduced (Alzahrani and Bulusu, 2020). Attackers deliberately present invalid transaction and bribe the dishonest nodes to vote it as valid. This can majorly diminish other node's trust in the blockchain. Our protocol mitigates this by introducing a randomised multiple node selections. To test this, we can model this attack with the malicious proposing node communicating off-chain with $\frac{1}{3}$ corrupted nodes and bribe them.

Another problem is the food supply chain's poor ability in early fraud detection (Flari et al., 2014). This is closely related to the inadequate food authenticity checks (Hong et al., 2017). Commonly these checks only involve product identification like barcode and Radio-frequency identification (RFID), but further analysis into food composition is required to identify adulterated products so as to serve as an important means to assure food safety. As a result, we aim to explore the use of sensor technology in better representing the physical product state in the future.

ACKNOWLEDGEMENTS

This work was funded by EPSRC (EP/S028366/1).

REFERENCES

- Alzahrani, N. and Bulusu, N. (2020). A new product anti-counterfeiting blockchain using a truly decentralized dynamic consensus protocol. *Concurrency and Computation: Practice and Experience*, 32(12):e5232.
- Buchnik, Y. and Friedman, R. (2019). Fireledger: A high throughput blockchain consensus protocol. *arXiv preprint arXiv:1901.03279*.
- Deirmontzoglou, E., Papakyriakopoulos, G., and Patsakis, C. (2019). A survey on long-range attacks for proof of stake protocols. *IEEE Access*, 7:28712–28725.
- Fadhel, N., Aniello, L., Margheri, A., Lombardi, F., and Sassone, V. (2019). Towards a semantic modelling for threat analysis of iot applications: a case study on transactive energy.
- Flari, V., Hussein, M., Maeder, R., Hubar, B., Marvin, H., and Neslo, R. (2014). Report on analysis of historical cases of food fraud. *Ensuring the Integrity of the European food chain*.
- Hammi, M. T., Hammi, B., Bellot, P., and Serhrouchni, A. (2018). Bubbles of trust: A decentralized blockchain-based authentication system for iot. *Computers & Security*, 78:126 – 142.
- Hong, E., Lee, S. Y., Jeong, J. Y., Park, J. M., Kim, B. H., Kwon, K., and Chun, H. S. (2017). Modern analytical methods for the detection of food fraud and adulteration by food category. *Journal of the Science of Food and Agriculture*, 97(12):3877–3896.
- Kamath, R. (2018). Food traceability on blockchain: Walmart's pork and mango pilots with ibm. *The Journal of the British Blockchain Association*, 1(1):3712.
- Kamble, S. S., Gunasekaran, A., and Sharma, R. (2020). Modeling the blockchain enabled traceability in agriculture supply chain. *International Journal of Information Management*, 52:101967.
- Litke, A., Anagnostopoulos, D., and Varvarigou, T. (2019). Blockchains for supply chain management: Architectural elements and challenges towards a global scale deployment. *Logistics*.
- Ojo, O. O., Shah, S., Coutroubis, A., Jiménez, M. T., and Munoz Ocana, Y. (2018). Potential impact of industry 4.0 in sustainable food supply chain environment. In *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, pages 172–177.
- Shevchuk, A. (2019). Traceability technology: fruits and vegetables trader case study. In *International Conference on Digital Technologies in Logistics and Infrastructure (ICDTLI 2019)*. Atlantis Press.
- Verhoeven, P., Sinn, F., and Herden, T. T. (2018). Examples from blockchain implementations in logistics and supply chain management: Exploring the mindful use of a new technology. *Logistics*.